

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №10
дисциплины «Основы программной инженерии»

Выполнил:
Матвеев Александр Иванович
2 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка и
сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: Работа с множествами в языке Python

Цель работы: приобретение навыков по работе с множествами при написании программ с помощью языка программирования Python версии 3.x.

Ход работы.


1. Создание нового репозитория с лицензией MIT.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Required fields are marked with an asterisk ().*

Owner *

 SashkaHacker ▾


Repository name *

laba10


 laba10 is available.

Great repository names are short and memorable. Need inspiration? How about [fluffy-guide](#) ?

Description (optional)

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

Initialize this repository with:

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore


.gitignore template: Python ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: MIT License ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

 You are creating a public repository in your personal account.

Create repository

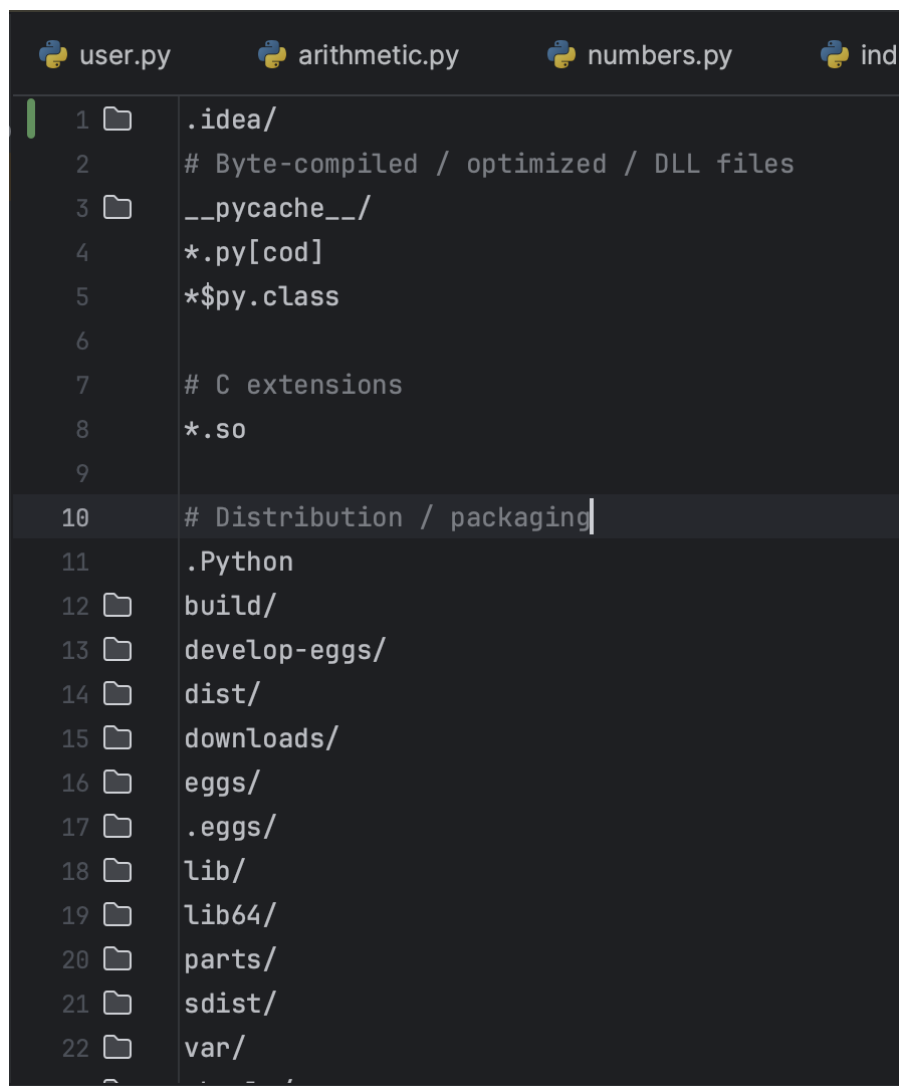
Рисунок 1 – Создание репозитория

2. Клонировал репозиторий на рабочий ПК.

```
Sashka@DESKTOP-U4RPSBI MINGW64 ~/Documents/GitHub
$ git clone https://github.com/SashkaHacker/laba10.git
Cloning into 'laba10'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
```

Рисунок 2 – Клонирование репозитория

3. Дополнил файл .gitignore необходимыми инструкциями.

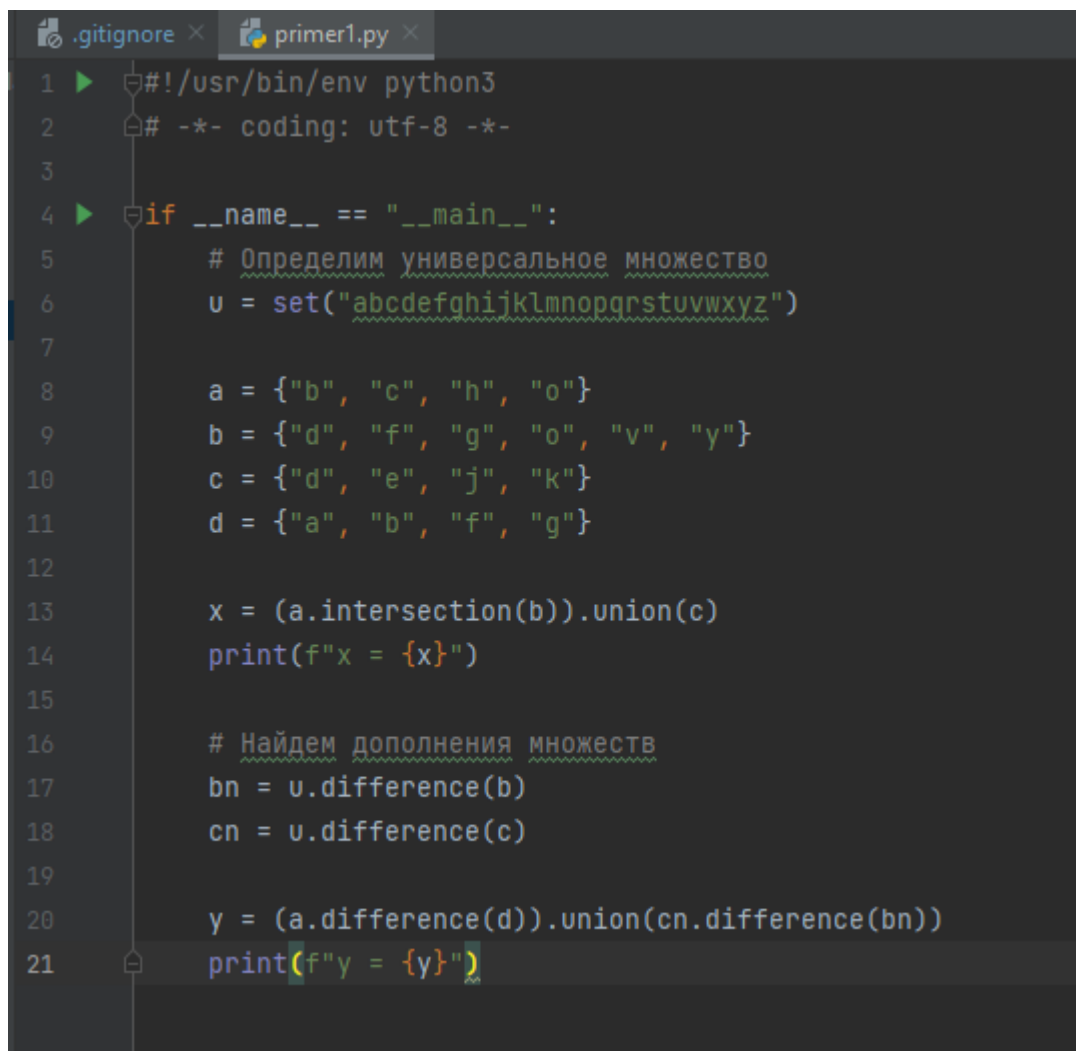


The image shows a code editor window with a dark theme. At the top, there are four tabs: 'user.py', 'arithmetic.py', 'numbers.py', and 'ind'. The 'user.py' tab is active. The editor displays the content of a '.gitignore' file. The text is as follows:

```
1  .idea/
2  # Byte-compiled / optimized / DLL files
3  __pycache__/
4  *.py[cod]
5  *$py.class
6
7  # C extensions
8  *.so
9
10 # Distribution / packaging
11 .Python
12 build/
13 develop-eggs/
14 dist/
15 downloads/
16 eggs/
17 .eggs/
18 lib/
19 lib64/
20 parts/
21 sdist/
22 var/
```

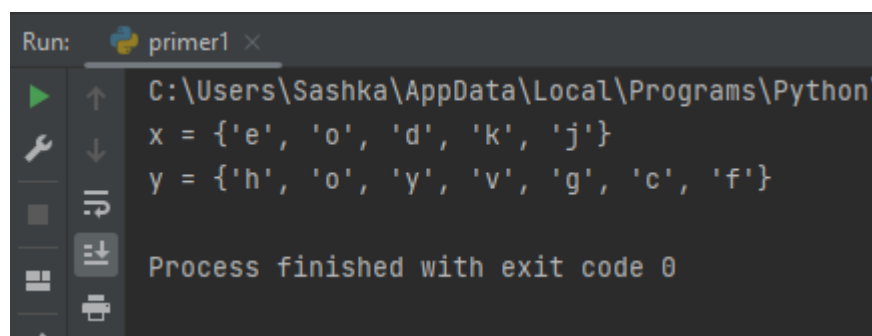
Рисунок 3 – Файл .gitignore

4. Проработка задания №1 из методических указаний.



```
1  > #!/usr/bin/env python3
2  > #- coding: utf-8 -*-
3
4  > if __name__ == "__main__":
5      # Определим универсальное множество
6      u = set("abcdefghijklmnopqrstuvwxyz")
7
8      a = {"b", "c", "h", "o"}
9      b = {"d", "f", "g", "o", "v", "y"}
10     c = {"d", "e", "j", "k"}
11     d = {"a", "b", "f", "g"}
12
13     x = (a.intersection(b)).union(c)
14     print(f"x = {x}")
15
16     # Найдем дополнения множеств
17     bn = u.difference(b)
18     cn = u.difference(c)
19
20     y = (a.difference(d)).union(cn.difference(bn))
21     print(f"y = {y}")
```

Рисунок 4 – Код программы

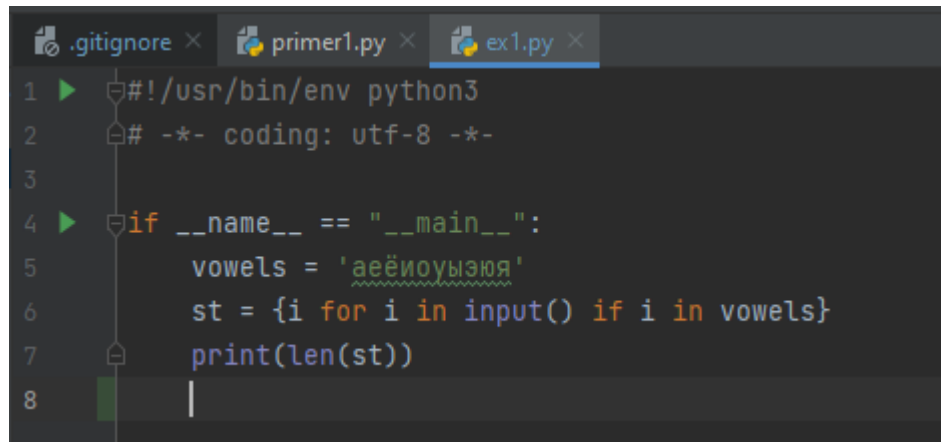


```
Run: primer1 x
C:\Users\Sashka\AppData\Local\Programs\Python
x = {'e', 'o', 'd', 'k', 'j'}
y = {'h', 'o', 'y', 'v', 'g', 'c', 'f'}
Process finished with exit code 0
```

Рисунок 5 – Работа программы

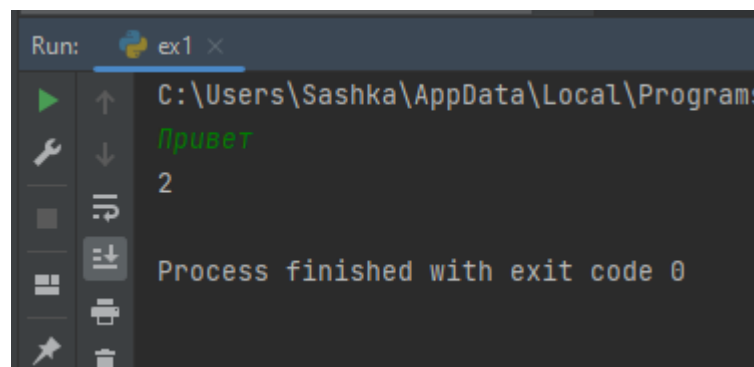
5. Выполнение задания №1.

Условие: подсчитайте количество гласных в строке, введенной с клавиатуры с использованием множеств.



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == "__main__":
5     vowels = 'аеёиоуыэюя'
6     st = {i for i in input() if i in vowels}
7     print(len(st))
8
```

Рисунок 6 – Код программы



```
Run: ex1 x
C:\Users\Sashka\AppData\Local\Programs\Python\Python39\python.exe
Привет
2
Process finished with exit code 0
```

Рисунок 7 – Пример выполнения программы

6. Выполнение задания №2.

Условие задания: определите общие символы в двух строках, введенных с клавиатуры.

```
.gitignore x primer1.py x ex1.py x ex2.py x
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == "__main__":
5     str1 = set(input())
6     str2 = set(input())
7     print(len(str1.intersection(str2)))
```

Рисунок 8 – Код программы

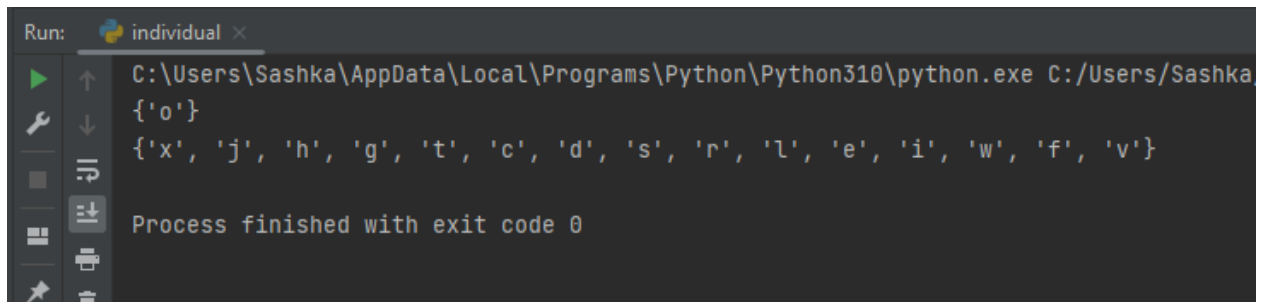
```
Run: ex2 x
C:\Users\Sashka\AppData\Local\Programs\Python\Python39\python.exe
Привет
Привет, как дела?
6
Process finished with exit code 0
```

Рисунок 9 – Пример выполнения программы

7. Выполнение индивидуального задания.

```
.gitignore x primer1.py x individual.py x ex1.py x ex2.py x
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == "__main__":
5     a = {'b', 'k', 'n', 'o', 'q'}
6     b = {'a', 'b', 'k', 'u'}
7     c = {'o', 'p'}
8     d = {'a', 'm', 'n', 'y', 'z'}
9     u = set("abcdefghijklmnopqrstuvwxyz")
10    x = (a | b) & c
11    y = (u.difference(a)) & (u.difference(b)).difference(c | d)
12    print(x)
13    print(y)
14
```

Рисунок 10 – Код программы



```
Run: individual x
C:\Users\Sashka\AppData\Local\Programs\Python\Python310\python.exe C:/Users/Sashka
{'o'}
{'x', 'j', 'h', 'g', 't', 'c', 'd', 's', 'r', 'l', 'e', 'i', 'w', 'f', 'v'}
Process finished with exit code 0
```

Рисунок 11 – Результат выполнения программы

Контрольные вопросы:

1. Что такое множества в языке Python?

О: множества – это структура данных, которая хранит уникальные элементы в неупорядоченном виде.

2. Как осуществляется создание множеств в Python?

О: `a = { }` или `a = set()` или `a = {i for i in range(10)}`

3. Как проверить присутствие/отсутствие элемента в множестве?

О: при помощи оператора `in` / `not in`

4. Как выполнить перебор элементов множества?

О: при помощи цикла `for`.

5. Что такое `set comprehension`?

О: синтаксическая конструкция, позволяющая создавать множества на основе существующих итерируемых объектов.

6. Как выполнить добавление элемента во множество?

О: `set.add()`

7. Как выполнить удаление одного или всех элементов множества?

О: `set.pop()` – удаляет рандомный элемент и возвращает его; `set.discard()` – удаляет элемент из мн-ва, при этом не будет ошибки, если такого элемента не существует; `set.remove()` – удаляет определенный элемент из мн-ва, вызовет ошибку, если элемента не существует; `set.clear()` – удаление всех элементов множества.

8. Как выполняются основные операции над множествами: объединение, пересечение, разность?

О: `set1.intersection(set2)` / `(&)` – пересечение; `set1.difference(set2)` / `(-)` – разность; `set1.union(set2)` / `(|)` – объединение.

9. Как определить, что некоторое множество является надмножеством или подмножеством другого множества?

О: `set1.issubset(set2)` – является ли `set1` подмножеством `set2`, `set1.issuperset(set2)` – является ли `set1` надмножеством `set2`.

10. Каково назначение множеств `frozenset`?

О: создание неизменяемого множества

11. Как осуществляется преобразование множеств в строку, список, словарь?

О: при помощи функций: `str()`, `list()` и `dict()`